Assignee: Intel Corporation
Docket No.: 2207/7942

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

| | | |
|---|---|---|
| APPLICANTS | : | Deborah T. Marr, et al. |
| SERIAL NO. | : | 09/490,172 |
| FILED | : | January 22, 2000 |
| FOR | : | ESTABLISHING THREAD PRIORITY IN A PROCESSOR OR THE LIKE |
| GROUP ART UNIT | : | 2171 |
| EXAMINER | : | Susan (Te Y.) Chen |

M/S: APPEAL BRIEF - PATENT
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

**ATTENTION: Board of Patent Appeals and Interferences**

**RESPONSE TO NOTICE OF NON-COMPLIANT APPEAL BRIEF**

Dear Sir:

The following is submitted in response to the Notice of Non-Compliant Appeal Brief dated July 6, 2007.

It is noted that Applicants have already filed a Reply Brief on July 19, 2007.

Changes to Section 3 of the Appeal Brief appear on page 2.

Changes to Section 5 of the Appeal Brief appear on page 3.

Amendments to the presentation of the claims with their status is presented in the Appendix. It is noted that Section 3 of the Appeal Brief as filed identified the status of the claims.

Please change Section 3 to read as follows:

**3. STATUS OF THE CLAIMS**

This application currently contains claims 1, 3-11, and 13-21. Claims 1, 3-11 and 13-21 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,105,127 to Kimura et al. in view of U.S. Patent No. 5,944,809 to Olarig et al. Claims 2 and 12 are cancelled.

Please change Section 5 to read as follows:

## 5. SUMMARY OF THE CLAIMED SUBJECT MATTER

Embodiments of the present invention pertain to establishing priority of a thread in a multi-threaded processor. A counter may be provided to store a value that assists in designating an amount of time that one thread will have priority over others. The value stored in the counter may be selected for each thread so as to give some threads a greater amount of priority (e.g., in access to a resource) compared to others.

In the embodiment of claim 1, a method is provided including assigning a value in memory to indicate which of a plurality of threads executed by a single processor has a higher priority (see, e.g., pg. 6, lines 4-6 and elements 3 and 4 in Fig. 1). In a next operation, a resource is allocated between the plurality of threads depending on a priority assigned to each thread (see, e.g., pg. 8, lines 14-21; an example of a resource is shown in Fig. 4 (element 81) and pg. 9, lines 6-19). In the allocation, a counter is provided with a predetermined value for the plurality of threads (see, e.g., Fig. 3, element 68), the value being selected by control logic depending on the priority assigned to each thread (see, e.g., pg. 7, line 21 to pg. 8, line 15).

In the embodiment of claim 10 a method is provided including assigning a value in an APIC TPR (Advanced Programmable Interrupt Controller – Task Priority Register) register for a thread via execution of operating system code to indicate which of a plurality of threads executed by said single processor has a higher priority (see, e.g., elements 3 and 4 and pg. 6, lines 4-22). In a next operation a resource is allocated between the plurality of threads depending on a priority assigned to each thread (see, e.g., pg. 8, lines 14-21; an example of a resource is shown in Fig. 4 (element 81) and pg. 9, lines 6-19). In the allocation, a counter is provided with a predetermined value for said plurality of threads (see, e.g., Fig. 3, element 68), the value being

selected by control logic depending on the priority assigned to each thread, said counter being used in said allocating operation (see, e.g., pg. 7, line 21 to pg. 8, line 15).

In the embodiment of claim 11, an apparatus is provided comprising a memory to store a value to indicate which of a plurality of threads to be executed by a single processor has a higher priority (see, e.g., pg. 6, lines 4-6 and elements 3 and 4 in Fig. 1). A resource is allocated between the plurality of threads depending on a priority assigned to each thread in said memory (see, e.g., pg. 8, lines 14-21; an example of a resource is shown in Fig. 4 (element 81) and pg. 9, lines 6-19). A counter loaded with a predetermined value by control logic for each thread in the memory depending on the priority assigned (see, e.g., Fig. 3, element 68). The counter is used to allocate said resource between said plurality of threads (see, e.g., pg. 7, line 21 to pg. 8, line 15).

In the embodiment of claim 20, an apparatus is provided for establishing thread priority in a single processor. The apparatus includes an APIC TPR register for a thread wherein execution of operating system code causes a value to be stored in said register to indicate which of a plurality of threads to be executed by the single processor has a higher priority (see, e.g., elements 3 and 4 and pg. 6, lines 4-22). A resource allocated between said plurality of threads depending on a priority assigned to each thread in said memory (see, e.g., pg. 8, lines 14-21; an example of a resource is shown in Fig. 4 (element 81) and pg. 9, lines 6-19). A counter loaded with a predetermined value by control logic for each thread in said memory (see, e.g., Fig. 3, element 68), said value being selected depending on the priority assigned, said counter being used to allocate said resource between said plurality of threads (see, e.g., pg. 7, line 21 to pg. 8, line 15).

## CONCLUSION

Appellants respectfully request that the Board of Patent Appeals and Interferences reverse the Examiner's decision rejecting claims 1, 3-11, and 13-21 under 35 U.S.C. § 103(a) direct the Examiner to pass the case to issue.

The Commissioner is hereby authorized to charge any fees which may be necessary for consideration of this paper to Kenyon & Kenyon Deposit Account No. 11-0600. A copy of this sheet is enclosed for that purpose.

Respectfully submitted,

Date: September 6, 2007

Shawn W. O'Dowd
(Reg. # 34,687)

KENYON & KENYON LLP
1500 K Street, NW
Suite 700
Washington, DC 20005
(202) 220-4200 telephone
(202) 220-42501 facsimile
DC1-670906

**APPENDIX**

(Brief of Appellants D. Marr et al.
U.S. Patent Application Serial No. 09/490,172)

## 8. CLAIMS ON APPEAL

The claims in their current form (including those claims under appeal) are presented below with their current status:

1. (Rejected) A method comprising:

assigning a value in memory to indicate which of a plurality of threads executed by a single processor has a higher priority;

allocating a resource between said plurality of threads depending on a priority assigned to each thread; and

providing a counter with a predetermined value for said plurality of threads, said value being selected by control logic depending on the priority assigned to each thread, said counter being used in said allocating operation.

2. (Canceled)

3. (Rejected) The method of claim 1 wherein in said allocating step, a first thread is given greater access to the resource than other threads when said first thread is assigned a higher priority than said other threads.

4. (Rejected) The method of claim 1, wherein in said allocating step, the other threads are given greater access to the resource than the first thread when said first thread is assigned a higher priority than the other threads and is not using said resource.

5. (Rejected) The method of claim 3 wherein said resource is a unit in a processor system.

6. (Rejected) The method of claim 5 wherein said resource is a decode unit.

7. (Rejected) The method of claim 6 further comprising:

providing instructions from a first thread to a first queue;

providing instructions from a second thread to a second queue;

supplying a first number of instructions to said decode unit from said first queue;

supplying a second number of instructions to said decode unit from said second queue;

selecting said first and second numbers based on said value in memory.

8. (Rejected) The method of claim 3 wherein said resource is a bus.

9. (Rejected) The method of claim 8 further comprising:

providing bus requests from a first thread to a first queue;

providing bus requests from a second thread to a second queue;

servicing a first number of bus requests from the first queue;

servicing a second number of bus requests from said second queue; and

selecting said first and second numbers based on said value in memory.

10. (Rejected) A method comprising:

assigning a value in an APIC TPR register for a thread via execution of operating system code to indicate which of a plurality of threads executed by a single processor has a higher priority;

allocating a resource between said plurality of threads depending on a priority assigned to each thread; and

providing a counter with a predetermined value for said plurality of threads, said value being selected by control logic depending on the priority assigned to each thread, said counter being used in said allocating operation.

11. (Rejected) An apparatus comprising:

a memory to store a value to indicate which of a plurality of threads to be executed by a single processor has a higher priority;

a resource allocated between said plurality of threads depending on a priority assigned to each thread in said memory; and

a counter loaded with a predetermined value by control logic for each thread in said memory depending on the priority assigned, said counter being used to allocate said resource between said plurality of threads.

12. (Canceled)

13. (Rejected) The apparatus of claim 11 wherein a first thread is given greater access to the resource than other threads when said first thread is assigned a higher priority than said other threads.

14. (Rejected) The apparatus of claim 11 wherein the other threads are given greater access to the resource than the first thread when said first thread is assigned a higher priority than the other threads and is not using said resource.

15. (Rejected) The apparatus of claim 13 wherein said resource is a unit in a processor system.

16. (Rejected) The apparatus of claim 15 wherein said resource is a decode unit.

17. (Rejected) The apparatus of claim 16 further comprising:

a first queue to store instructions from a first thread;

a second queue to store instructions from a second thread;

control logic coupled to said first and second queues and said decode unit, said control logic to permit a first number of instructions to be transferred to said decode unit then a second number of instructions to be transferred to said decode unit, said first and second numbers being selected based on said value in memory.

18. (Rejected) The apparatus of claim 13 wherein said resource is a bus.

19. (Rejected) The apparatus of claim 18 further comprising:

a bus unit including

a first queue storing bus requests from a first thread;

a second queue storing bus requests from a second thread;

control logic coupled to said first and second queues, said control logic to control servicing of a first number of bus requests from the first queue and a second number of bus requests from said second queue, said first and second number being selected based on said value in memory.

20. (Rejected) An apparatus for establishing thread priority in a single processor comprising:

an APIC TPR register for a thread wherein execution of operating system code causes a value to be stored in said register to indicate which of a plurality of threads to be executed by said single processor has a higher priority;

a resource allocated between said plurality of threads depending on a priority assigned to each thread in said memory; and

a counter loaded with a predetermined value by control logic for each thread in said memory, said value being selected depending on the priority assigned, said counter being used to allocate said resource between said plurality of threads.

21. (Rejected) The apparatus of claim 20 wherein a first thread is given greater access to the resource than other threads when said first thread is assigned a higher priority than said other threads.